Chapitre II
Manipulation de documents XML

II.1- Xpath

Plan du cours

- Introduction
- XPath 1.0
- XPath 2.0
- Modèle de données xpath
- Expression simple de chemin
- Expression de chemin par axes nodaux
- conclusion

•Introduction

•XPath est un langage permettant de sélectionner des parties d'un document XML. Il est utilisé dans de nombreux dialectes XML

•Le langage XPath n'est pas un langage autonome. C'est un langage d'expressions utilisé au sein d'un autre langage hôte.

•

•XSLT et XQUERY fait usage intensif de XPath pour désigner les parties à traiter

Introduction

- La syntaxe de XPath n'est pas une syntaxe XML

- Le cœur de XPath est formé des expressions de chemins permettant de décrire des ensembles de nœuds d'un document XML

•XPath 1.0

•Le résultat d'une requête XPath est un ensemble de nœuds qu'il faut voir comme un ensemble de références vers des nœuds de l'arbre XML.

•Ensemble
- •non ordonné (en réalité on récupère les nœuds dans l'ordre du document)
- • sans doublon

XPath 2.0

•Le résultat d'une requête XPath est une séquence de nœuds ou de valeurs atomiques.

- séquence :
  - ordonné
  - doublons possibles

Modèle de données xpath

- Un document est vu comme un un arbre
- Sept sortes de noeuds :
  - 1. Document Node (le tout premier nœud, représentant le document même)
  - 2. Elemen Nod
  - 3. Attribute Node
  - 4. Text Node
  - 5. Comment Node
  - 6. ProcessingInstruction node
  - 7. Namespace Node

Modèle de données xpath

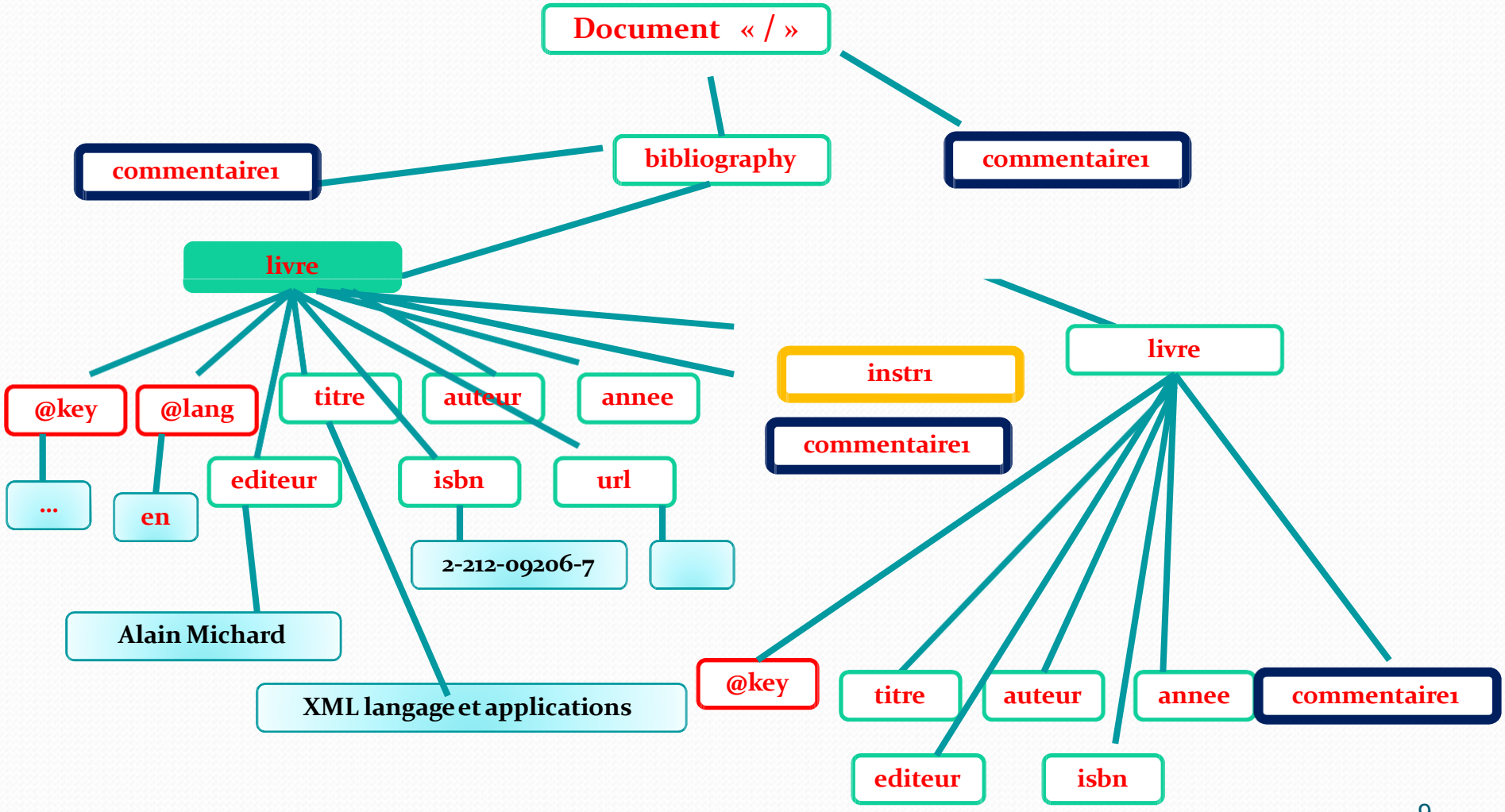
- Exemple

```
<?xml version="1.0" encoding="iso-8859-1"?>

<!-- Time-stamp: "bibliography.xml 3 Mar 2008
16:24:04" -->

<!DOCTYPE bibliography SYSTEM
"bibliography.dtd" >

<bibligraphy>
 <!– premier livre-->
<?inst1     ?>
<livre key="Michard01" lang="fr">
<titre>XML langage et applications</titre>
<auteur>Alain Michard</ auteur >
<annee>2001</ annee >
<editeur>Eyrolles</ editeur >
<isbn>2-212-09206-7</isbn>
<url>http://www.editions-
                    eyrolles/livres/michard/</url>
</ livre >
```

```
< livre key="Zeldman03" lang="en">
  <!– deuxieme livre-->
<titre>Designing with web standards</titre>
< auteur >Jeffrey Zeldman9</ auteur >
< annee >2003</ annee >
< editeur >New Riders</ editeur >
<isbn>0-7357-1201-8</isbn>

</ livre >
</bibliography>
```

Modèle de données xpath

- Exemple: extrait de l'arbre xpath



Expression simple de chemin

- **chemin**

- Un chemin est une suite de termes permettant d'atteindre un ou plusieurs élément depuis la racine du document sélectionnant un nœud ou ensemble de nœuds.

- Les termes sont séparés par " /"ou " / /".

Expression simple de chemin

Opérateur '/' de composition de chemins

1. permet de composer des expressions de cheminement pour créer de véritables chemins dans un arbre XML

2. Sa sémantique est proche du même opérateur '/' des chemins des fichier sous Windows:
   - C:/mi/informatique/ 2021/3L/SSd/tp1.pdf
   - C:/mi/informatique/ 2021/3L/SSd/*

Expression simple de chemin

Opérateur '/' de composition de chemins

1. Exprime une relation direct élément père- element fils.
2. Une expression de la forme : ***elem/obj*** sélectionne les objets fils directs de l'"èlement '***elem***' qui porte le nom '***obj***"
3. '/ ' au début de chemin désigne la racine de document (début dedocumentnonpasl'élementracine)
4. Cas particulier :
   - Le chemin '/ ' désigne la racine de document

Expression simple de chemin

Opérateur '/' de composition de chemins

1. **Exemples**:

expression	Eléments sélectionnés
/	la racine du document
/bibliography	les éléments 'bibliography' qui est le premier élément (élément racine)
/bibliography/livre	les éléments 'livres ' fils de l'element 'bibliography'
/bliography/titre	Rien

Expression de chemin

Opérateur '//' de composition de chemins

1. Exprime une relation indirecte élément père- éléments descendants.
2. Une expression de la forme **elem//obj** sélectionne les objets fils indirects(descendants) de l èlement '**elem** ' qui porte le nom '**obj**'
3. Remplace toute une partie de chemin
4. Cas particulier :'// ' au début de chemin designe plusieurs ancêtres depuis la racine de document

Expression de chemin

Opérateur '//' de composition de chemins

1. **Exemples**:

expression	Eléments sélectionnés	Chemin remplacé
//	erreur	
//livre	les éléments 'livre'	/bibliography//bibliography/
/bibliography//titre	les éléments 'titre' fils de l'element 'bibliography'	/livre/

Expression de chemin

Jocker '*'

- '* ' a pour exprimer un ensemble d'objets(elements ,attributs)quelque soit leurs noms

1. **Exemples**:

expression	Eléments sélectionnés
/biliograpy//*	Tous les éléments descendants de l'element 'bibliography'
//livre /*	Tous les elements files de l'element 'livre'

Expression de chemin

Designer un attribut , le jocker ‘@ ’

- ‘@ ‘ a pour designer que le type est un attribut

1. **Exemples**:

expression	Eléments sélectionnés
//livre/@key	L’attribut ‘key ‘ de l’element ‘livre’
/bibliography//@*	Tous les attributs descendants de l’element ‘bibliography’
//@*	Tous les attributs

Expression de chemin

Designer les commentaires :comment()

- Comment() permet de sélectionner les commentaires

1. **Exemples**:

expression	Eléments sélectionnés
//livre/comment()	L'attribut 'key ' de l'element 'livre'
/comment()[position()=1]	Le premier commentaire dans le document

Expression de chemin

Designer les instructions: *processing-instruction()*

- permet de sélectionner les instructions de traitement

1. **Exemples**:

expression	Eléments sélectionnés
//processing-instruction()	Les instructions de traitement

Expression de chemin

Designer les nœuds texte :texte()

- permet de sélectionner les fragment texte : valeurs des éléments , tabulations, espaces …

1. **Exemples**:

expression	Eléments sélectionnés
//livre/texte()	Les valeurs des elements + tabulations dans l'element livre
//texte()	Tous les fragments texte

Expression de chemin

Designer les instructions: *node()*

- permet de sélectionner tout type de nœud

1. **Exemples**:

expression	Eléments sélectionnés
//node()	Tous les elements , attributs,commentaires …

Expression de chemin

- **Les prédicats**

- Permettent d'ajouter une condition sur les éléments ou attribut recherchés
- **Syntaxe** :Elem[predicat]
- **Exemples**:

expression	Eléments sélectionnés
//livre[url]	Les livre ayant un element fils 'url'
//livre[not url]	Les livre n'ayant aucun element fils 'url'
//livre[@key]	Les livre ayant un attribut ' key'
//livre[note @key]	Les livre n'ayant pas l'attribut ' key'
/biliograpy/livre[@key='Michardo1']	Tous les éléments ayant la valeur Michardo1 dans l'attribut 'key'
//livre [count(auteur)=2]	Tous les éléments livre ayant deux éléments fils 'auteur'
//livre [count(@*)=2]	Tous les éléments livre ayant deux attributs

Expression de chemin

- **Les prédicats**
  - **operateurs:**

Opérateur	Type opérandes
and, or et not()	logiques
=, !=, , <=, >=	nombres, des chaines de caractères
+, -, *, div, mod	nombres

Expression de chemin

- **Les prédicats**
  - **Fonctions arithmétiques :**

Opérateur	Type opérandes
Floor(nbr)	l'entier immédiatement inférieur à sa gauche.
ceiling(nbr)	l'entier immédiatement supérieur à sa droite.
rounde(nbr)	le nombre entier le plus proche de l'argument

- **exemples**:

chemin	elements
floor(4.2)	4
floor(-4.2)	-5
ceiling(4.2)	5
ceiling(-4.2)	-4
round(4.2)	4
round(4.6)	5

Expression de chemin

LES PRÉDICATS

- **Fonctions chaines de caractères**:

Fonction	Role
Concat(chaine1,chaine2)	Elle retourne le résultat de la concaténation des arguments
substring(*chaine* , *pos* , *long*)	retourne la sous-chaîne commençant de le position 'pos' d'une longueur 'long'
string-length(*chaine)*:	*retourne* ***la longueur de la chaine 'chaine'***
normalize-space(chaine)	Retourne la chaine 'chaine ' normallisée

- **exemples**:

chemin	elements
concat(« AB »,« CDE »,« EF »)	ABCDEF
string-length("système "):	7
substring(" 12345",2,3) .	234
normalize-space(" c'est un cours facile ")	**c'est un cours facile ")**

Expression de chemin

LES PRÉDICATS

- **Fonctions chaine de caractères**:

Fonction	Role
starts-with(chaine1,chaine2)	Boolean , teste si chaine1 commence par chaine2
contains(chaine1,chaine2)	Boolean , teste si chaine2 est sous chaine de chaine1

- **exemples**:

chemin	resultat
starts-with(”informatique “ , “form”)	false
contains(”informatique “ , “form”)	true

Expression de chemin

LES PRÉDICATS

- **Fonctions xpath**:

Fonction	Role
position ()	numéro d'ordre du nœud courant dans la liste de nœuds contextuelle
last()	position du dernier nœud dans le contexte courant
count(elem) :	nombre de nœuds 'elem' du nœud courant

- **exemples**:

chemin	elements
//livre[position ()=1] ou bien //livre[1]	Le premier élément 'livre'
//livre[position ()=last()]	Le dernier élément 'livre'
//livre[count(@*)=2]	nombre de nœuds 'elem' du nœud courant
//*[count(@*)=1 and count(*)>2]	Tous les éléments ayant un seul attribut et plus de deux éléments fils

Expression de chemin par axes nodaux

- XPath permet au travers de la notion d'axes nodaux, de référencer des sous-arbres complexes type frères, parent, ….

- La Syntaxe générale d'un chemin

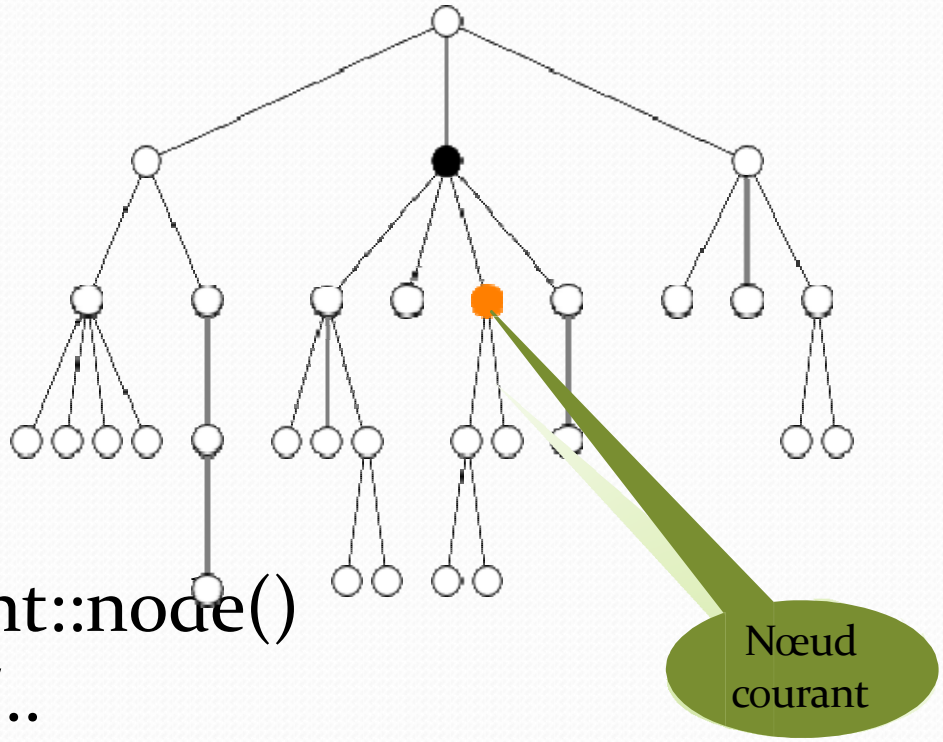
  axe::filtre[predicat1][predicat2]…

- exemple:

/bibliography//titre
/ancestor::titre[position()=2]
Sélectionne le deuxième element du type étudiant ancetres du nœud courant 'titre'

Expression de chemin par axes nodaux

- Axes
  - Parent
    - Le parent

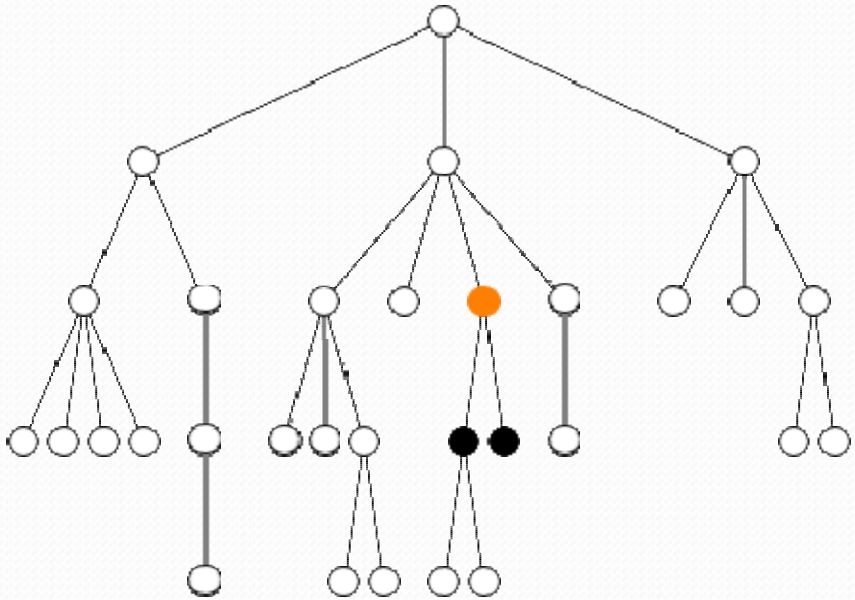


Exemple :
//bibliography/livre/parent::node()
//bibliography/livre/titre/..
- Equivalent à
//bibliography/livre

Expression de chemin par axes nodaux

- Axes
  - Child :par défaut

    Y Enfants directs

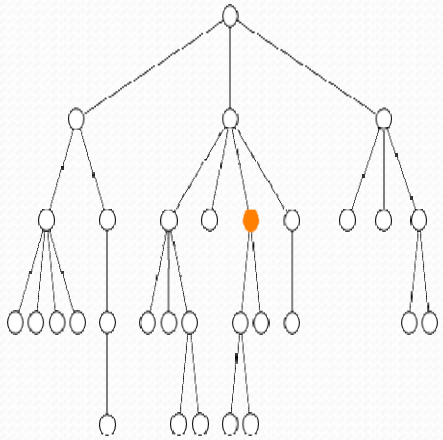


Exemple :

//livre/child::* équivalent à //livre/*

Expression de chemin par axes nodaux

- Axes
- Self abréviation '.'
  - Le nœud lui-même (égalité)



Exemple :

//bibliography/self::*

- Equivalent à

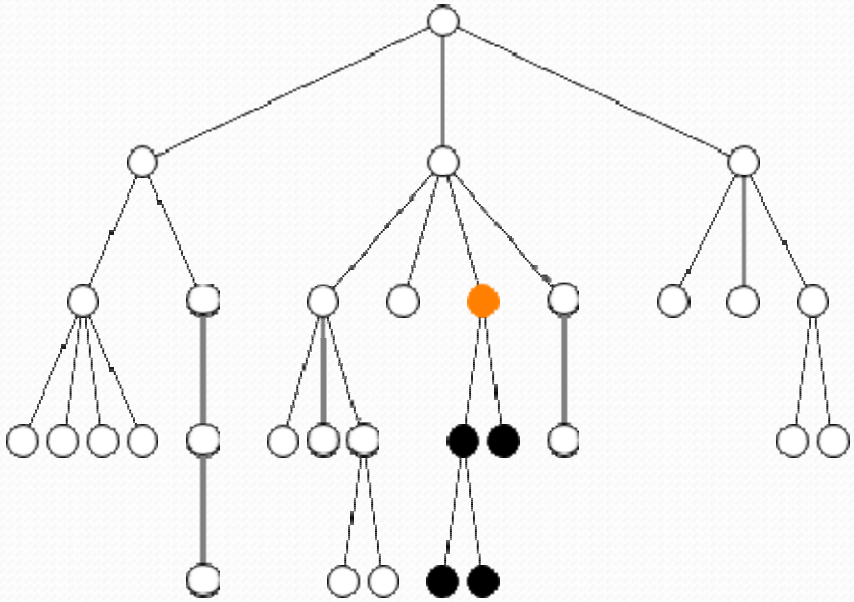
//bibliography/*

- Equivalent à

//bibliography/./*

Expression de chemin par axes nodaux

- Axes
  - descendant
    - Descendant strict



- Exemple :

/bibliography/descendant::*

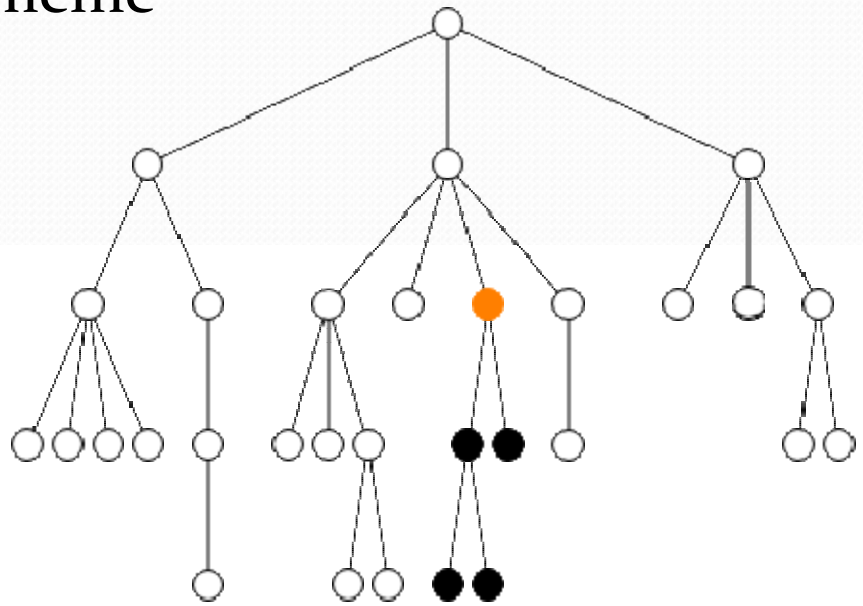
- Equivalent à

/bibliography//*

Expression de chemin par axes nodaux

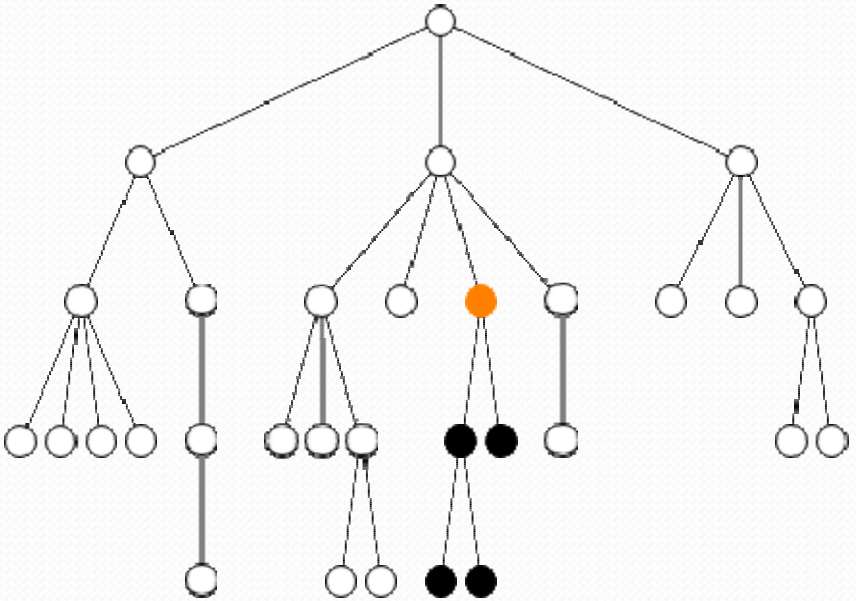
- Axes
  - descendant-or-self

    Y Descendant ou le nœud lui-même



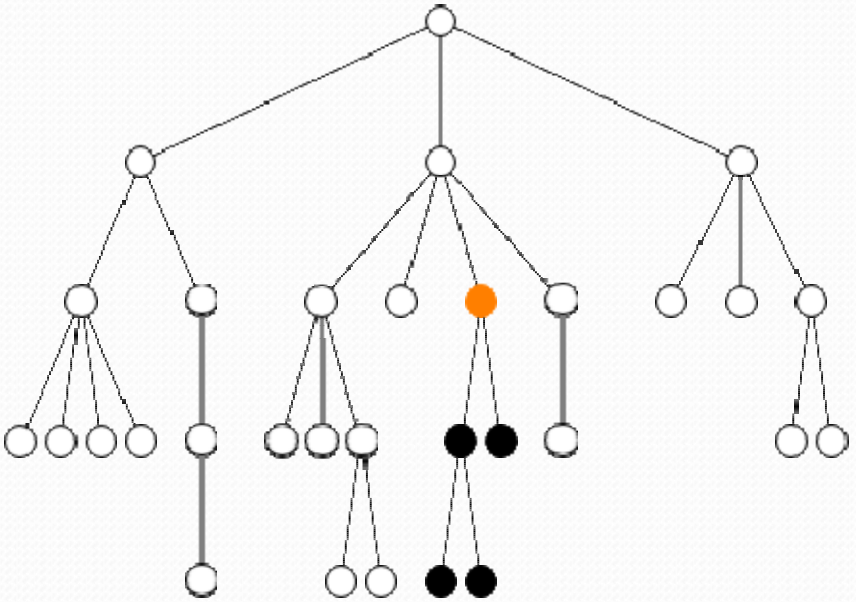
Expression de chemin par axes nodaux

- Axes
- ancestor
  - Y A ncêtres strict



Expression de chemin par axes nodaux

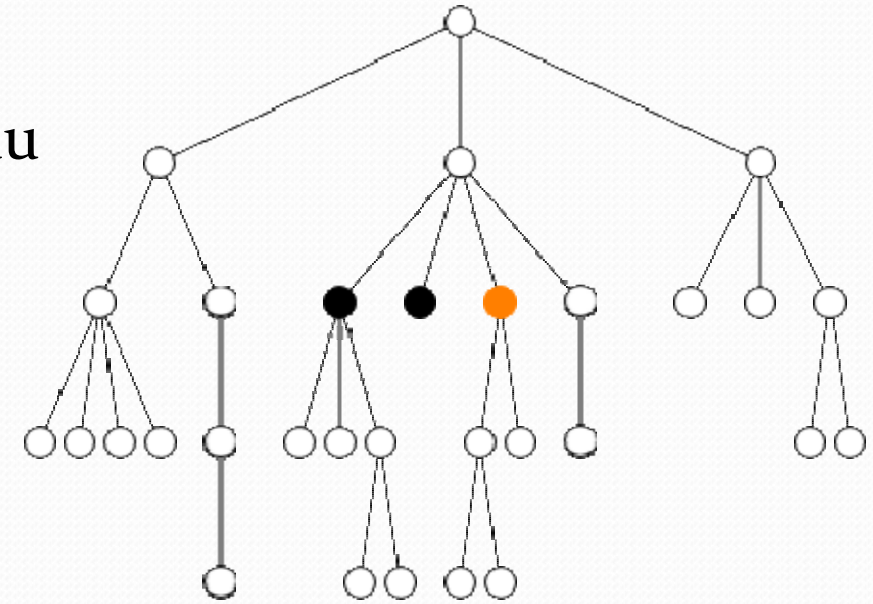
- Axes
  - ancestor-or-self
    - Ancêtres ou le nœud lui-même



Expression de chemin par axes nodaux

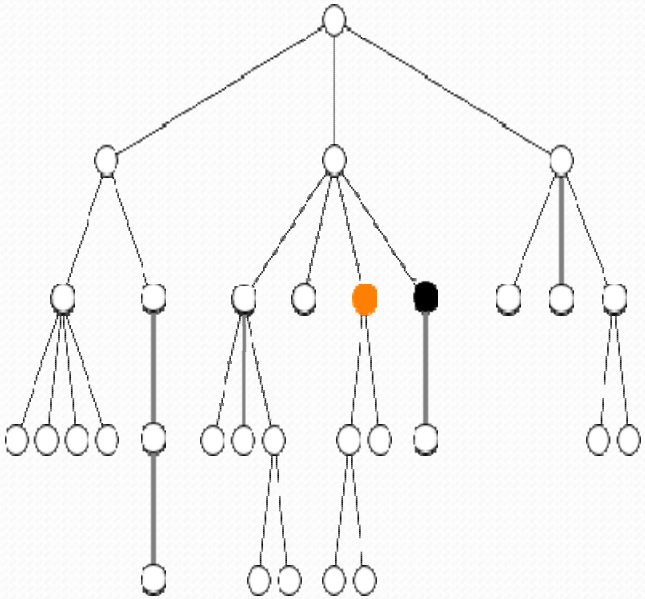
- Axes
  - preceding-sibling

    Y Frère gauche (enfant du même parent)



Expression de chemin par axes nodaux

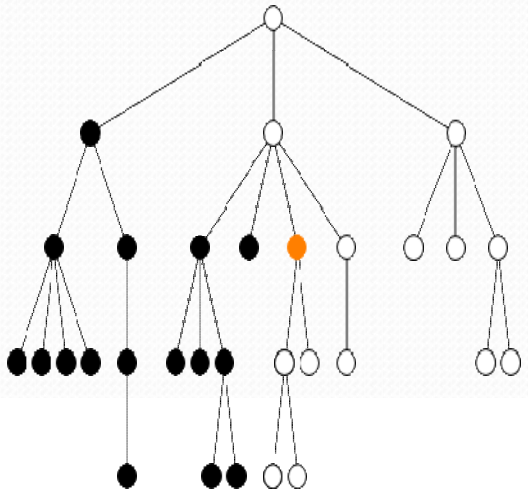
- Axes
  - following-sibling
    - Frère droit (enfant du même parent)



Expression de chemin par axes nodaux

- Axes
  - preceding

Y À gauche



Expression de chemin par axes nodaux

- Axes
  - following Y doite